ZX SPECTRUM FILES
microhobby.speccy.cz/zxsf/pagina_1.htm



Nueva etapa, nuevos retos

Novedades en software

Comentarios

Patas arriba: Zorro (Datasoft)

Curso de BASIC compilado (III)

Volvemos en una nueva etapa, con ganas e ilusión. Ayudadnos en lo que buenamente podáis, enviando cualquier colaboración a la siguiente dirección e-mail.



rdisky-nospam-@hotmail.com

EDITORIAL

Ya era hora, por fin, de que la revista siga saliendo, aunque esta vez haya sido tras un compás de varios meses, tiempo durante el cual he tenido que reflexionar mucho no solo en cuanto a la calidad de la misma, sino sobre el formato. Cuando saqué el número anterior, inmediatamente después di curso una encuesta por medio de la cual os preguntaba sobre el nuevo modelo de la revista. A la vista de los resultados, habiendo una práctica igualdad entre las opciones de continuar con el mismo modelo de antes (alargado) y el nuevo (apaisado), finalmente he decidido cambiar al que tenéis delante de vuestros ojos, para comodidad vuestra y exclusivamente pensado para leerlo directamente desde vuestros PC's.

Las fuentes true-type no serán tantas y procuraré usar las justas y al tamaño adecuado. En los foros podréis luego comentar qué os parece y si véis alguna posibilidad de mejora, estaré abierto a vuestras sugerencias, pero eso sí, la revista se tiene que ir amoldando poco a poco a vuestros gustos, con el voto mayoritario de sus lectores. Por otra parte, en medio del suspense generado por este largo compás de espera, ha sido esperar al mes de febrero cuando he decidido coger el toro por los cuernos y renovar por completo los contenidos y procurar que sean, al menos, de interés. Prácticamente, un año después del anterior número, vuelvo con ganas y espero que no se acabe aquí. Cargado de comentarios, espero que el presente ejemplar, al menos, os guste.

Y avanzo que el siguiente número saldrá más pronto que tarde, incluyendo un informe sobre lo acontecido en la RetroMadrid, con todas sus novedades. Y mucho más...

Un saludo a todos.

SUMARIO

Redacción y maquetación: Neil Parsons
Colaboraciones: Iván Sánchez, Tony Brazil, Mishazx y Na_th_an.
Portada: © Alfonso Azpiri.

Correo electrónico para consultas y sugerencias:
rdisky@hotmail.com

OPINIÓN

No me esperaba tomarme un descanso así de largo, pero aquí estoy. Y vuelvo con ganas, después de pasar una temporada algo turbia, entre mi actual situación laboral (ni falta que hace relatar cómo está el patio) y mi estado emocional, pero vuelvo con la fuerza necesaria de reconducir la revista por nuevos derroteros y solventar en buena medida mi falta de tanto tiempo sin tocar mi ZXSF.

Tras este periodo, ya véis que he renovado por completo la presentación de la revista y su formato, como ya estaba anunciado desde el año pasado. Un formato que muchos aplaudiréis con el debido entusiasmo, pero alguna crítica que otra también hace falta, no exagerar demasiado con vuestras efusivas, ni tampoco quiero que me tiréis tomates ni huevos podridos (bueno, si cae alguno, igual me lo mereceré... :D).

En el presente número no hay apenas gran cosa que destacar. Mejor lo valoréis vosotros mismos. Eso sí, tened en cuenta que me pasaré largo tiempo en preparar el siguiente número, si es que lo saco.

Por otra parte, el presente número es muy especial, pues lo he dedicado a los juegos de Spectrum en general. De fondo, tenéis un ejemplo de ello. Hay incluso un artículo dedicado al “Zorro” de Datasoft, juego que es muy fácil de acabar si se siguen las indicaciones dadas. Aparte, varios comentarios (muchos de ellos del propio IvanZX) de juegos tanto actuales como clásicos y reseñas a lo que acontecerá este sábado.

No falta a la cita Na_th_an y su curso de BASIC compilado, con la tercera entrega del mismo. La sección de arte gráfico continúa y lo hago exponiendo varias pantallas creadas por diferentes artistas para varios certámenes de renombre (p. ej.: Chaos Constructions 2004 demo party), muchas de ellas elaboradas con mucha imaginación y técnica, así como alguna que otra creación de las mías.

Y un inciso que quiero hacer, para todo aquél que quiera colaborar en la revista; les invito a proponer artículos, entrevistas, análisis de hardware y software, etc. lo que se les ocurra.

Finalmente, confío en que el siguiente número, saldrá cuando tenga que salir.

NEIL PARSONS

RetroMadrid

XVI edición



14 de marzo de 2009
Horario 10:30 a 19:30

Centro Cultural El Greco
C/El Greco s/n. Madrid

Entrada gratuita

http://www.retromadrid.es







La Corona Encantada

Karoshi Corp. en asociación con **Matra** nos pone en bandeja una superproducción digna de haber sido publicada en los mejores años del **ZX Spectrum** y **MSX**, pues son los sistemas para los que va destinado este juego. Con portada del siempre admirado **Alfonso Azpiri**, esta creación que en un principio apareció para los ordenadores MSX de cara al concurso de videojuegos **MSXdev'08** quedando en el tercer lugar de la clasificación habiendo sido premiado a la mejor música y mejor gameplay, aparece también para el ordenador de Sinclair. **El propio Azpiri firmó ejemplares en la reciente RetroMadrid '09**. Está disponible a la venta por estas cantidades:

19,95 (pack MSX)
14,95 (pack ZX)

El juego viene en pack de lujo, incluyendo su cinta/cartucho y un miniCD con extras.

http://www.matranet.net



Capitán Sevilla, el remake

Tras cerca de tres años de programación, desarrollo, betatesteo y todo lo que queráis echarle, ya tenemos por fin el deseado remake del conocido videojuego de Dinamic "**Capitán Sevilla**", realizado a la par entre Luis I. García Ventura (**Benway**) y Daniel Celemin (**LordFred**) para el sello CEZ GS (división de remakes) y el último en aparecer bajo el mismo, como ya descubriréis leyendo los foros de CEZ.

Todo un pelotazo, dicho con la mejor de las expresiones. El amplio despliegue y los pequeños detalles hacen de este remake el juego ideal para pasar largo rato jugando sin cansarte lo más mínimo. La introducción, en forma de cómic (fiel al publicado por el dibujante **Max** en la revista **Microhobby**) nos sumerge en la historia y lo que nos encontramos después en el juego, es mejor verlo por vosotros mismos. Lo dicho, *un pelotazo* y a buen seguro es el mejor juego remake realizado en mucho tiempo.

http://computeremuzone.com/ficha.php?id=754

Vuelve el sabor de los 80...

Compiler Software ha decidido embarcarse en la aventura de comercializar los juegos más destacados del momento y lo hace con el videojuego de **World XXI Soft** de reciente factura: **Escuela de Ladrones**, al precio de 8,75 Euros. Seguro que lo vale. **http://compiler.speccy.org/**

Mojon Twins se independizan

Los Mojon Twins (grupo formado por varios programadores ex-miembros de CEZGS) crean su propia web, en forma de blog, y a través de ella nos anuncian novedades a punto de caramelo, tanto para **ZX Spectrum** como para **Amstrad CPC**. Y eso no es todo, amenazan con extender sus dominios al **MSX**. Quedáis avisados... **http://www.mojontwins.com**

Próximas novedades en software

De momento, salvo dignas novedades referidas a lo largo de esta semana (ya tenemos lo nuevo de los Mojon Twins: **Phantomas Tales I**) se avecinan más cosas para este año, aunque de momento solo son comentarios y rumores. Algo más seguro, es lo próximo de **Topo sigloXXI**, del que no voy a desvelar su nombre, pero será una auténtica sorpresa para cuando salga. Discreción, ante todo. Conoceremos más detalles en adelante.

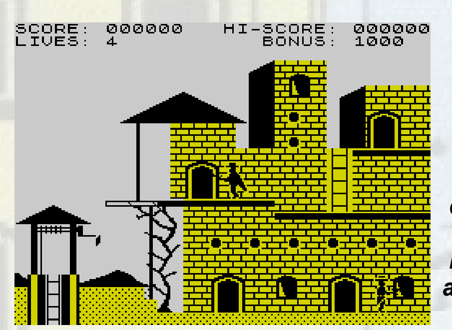
PATAS ARRIBA

Una nueva sección da comienzo, una vez que se inaugura esta nueva etapa de la revista, en la cual se tratará de destripar con todo lujo de detalles aquellos juegos tanto clásicos como actuales, para todo aquél que se haya quedado aplanado sin saber qué hacer para seguir adelante. Mientras que hay lugares en internet donde aplacar a base de videos, pokes y guías varias las ansias por lograr llegar al final de ese juego que tenías por favorito, aquí se tratará de hacer lo propio desde el papel, desde la parte escrita.

Lo hago con un juego sumamente sencillo de acabar, sin necesidad de pokes: Zorro de Datasoft. Hace algún tiempo, destiné un comentario sobre este juego en particular en mi blog "Program:Bytes:48k", y tuve la oportunidad de incluir en él un clip de YouTube con una guía visual de cómo acabarlo.

Pero si preferís la lectura a la mera visualización de esos archivos, entonces no dejo pasar más tiempo en veleidades sin sentido y os dejo ir directamente al meollo del asunto y aprovechar a desempolvar este juego.

Una vez que comenzamos a jugar, subimos directamente al pozo, encaramándonos primero por el árbol hasta la primera planta del edificio, nos dirigimos al extremo y desde allí saltamos limpiamente hasta el brocal para recoger el pañuelo y los primeros puntos en nuestra cuenta.



Bajar luego por el pozo hasta el nivel inferior, en el lago subterráneo. Saltando sobre las bolas flotantes, lograremos acceder a la plataforma superior, con cuidado de no caernos a un lado o a otro, y nos colgamos en la barra superior, empezamos a movenos lo justo como para caer sobre la tercera bola.

Desde allí, damos un nuevo salto hacia la derecha hasta el saliente inferior que nos conduce a una nueva pantalla, en el que vemos un complejo sistema de tracción y poleas y un primer objeto relampagueando a la vista: una copa. Lo primero que tenemos que hacer es ir directamente al piso inferior, entrando por la esquina inferior izquierda; en ella iremos a por una gran maceta. Una vez con ella en nuestras manos, volvemos arriba y nos dirigimos hasta donde se halla parada una piedra rodante. La movemos con solo rozarla y directamente nos subimos con ella en la primera plataforma. Soltamos también la maceta en el mismo lugar donde nos encontramos (en la misma plataforma) y vemos cómo la bola rueda hasta la siguiente plataforma y el peso tanto nuestro como de

la maceta hará el resto. La compuerta que tapaba el camino hasta la copa se abre y se quedará así hasta que pasemos a recogerla. Volvemos a la superficie por donde venimos, con cuidado de no resbalar y caer al lago. Ya en tierra firme, subimos o seguimos literalmente en el mismo nivel y nos dirigmos a la pantalla del sofá. En ella y en los próximos minutos, varios objetos a recoger y usar en diversos momentos. El primero de ellos es una botella. Saltamos sobre el sofá, haciéndolo justo enmedio del mueble, hasta la plataforma derecha y desde allí haremos un largo recorrido hasta poder acceder al objeto en cuestión, pero cogiendo antes una llave que haremos siempre que tengamos que recoger un objeto del mismo lugar.

Con la botella en nuestro poder, nos trasladamos a la taberna y se la damos al primero que se encuentra allí, medio tumbado sobre el mostrador. Se quedará "tarumba" enseguida. Nos montamos encima suya (sin hacer guarrerías) y saltando sobre él llegaremos hasta el piso superior de tres puertas. De vez en cuando salen guardias por alguna de ellas. Si conseguimos que uno de ellos salga por la puerta de la izquierda, procuraremos empujarlo en la lucha hasta hacerlo caer sobre la lámpara. Éste se agarrará a ella y su peso hará subir una caja que la sujetaba. Debajo de la caja hay una escalera. Bajamos inmediatamente por ella y ya tenemos en nuestro poder la copa.

Volvemos de nuevo a la pantalla del sofá, esta vez para recoger un hierro para marcar reses. Con el hierro, nos dirigimos a los establos y lo soltamos sobre el horno. Vamos hasta el fuelle, nos subimos en él y saltando haremos avivar el fuego. Podemos saltar un par de veces como poco.

El hierro se habrá calentado lo suficiente como para recogerlo y darle en todo el cachete al toro bravido que tenemos a su lado, saliendo escopeteado a otros andurriales. Ya tenemos el siguiente objeto en nuestro poder: una herradura.

Nuevo objeto a la vista: una trompeta, en la susodicha escena del sofá. Con ella nos dirigimos a la pantalla anterior izquierda y nos colocamos justo en el extremo de una especie de trampolín sito justo delante del edificio central. La hacemos sonar tal como en los tiempos de la "mili" (nos acordaremos de tocarla con firmeza) y un soldado saldrá a toda leche saltando al vacío. Tenemos que estar en el sitio justo para que en el impulso posterior podamos agarrarnos al saliente y desde allí saltar a la primera plataforma y subir hasta el torreón donde nos aguarda otro objeto: unas botas. Si os falta precisión, observar con detalle el vídeo colgado en mi blog y sabréis dónde tendréis que colocaros exactamente: es en la punta izquierda de la plataforma, no llegando a tocar el suelo.

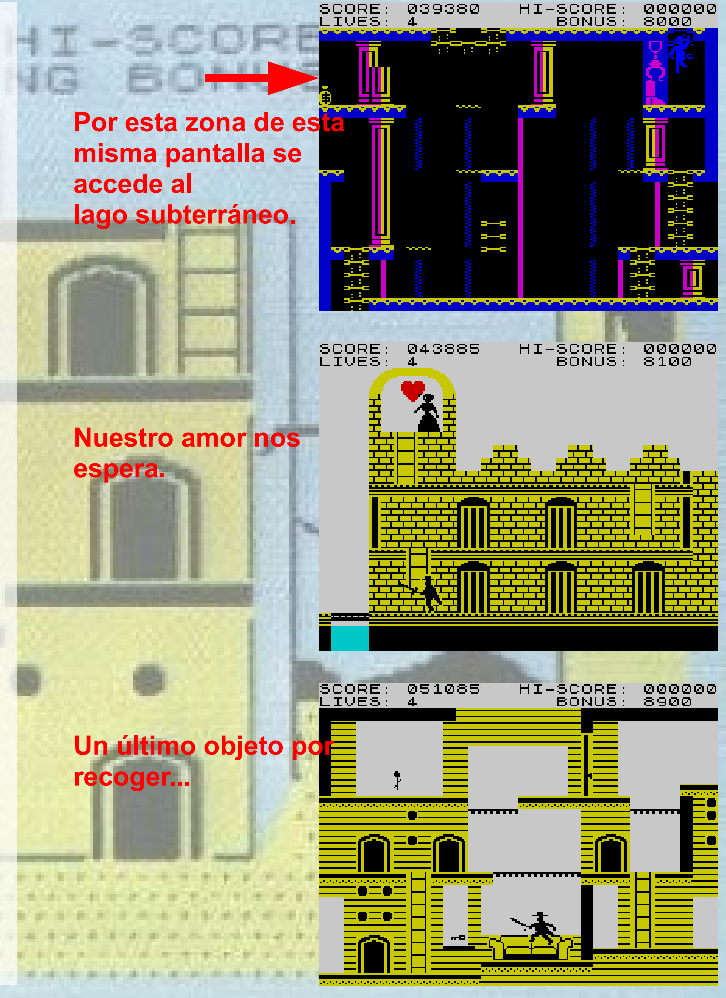
Con los tres objetos en nuestro poder y alcanzado muchas bonificaciones, tenemos el pase a la cárcel asegurado. Pero para llegar hasta allí, tenemos que atravesar las catacumbas. Para ello abriremos una tumba. La forma de hacerlo es la siguiente: como siempre, en la pantalla del sofá irán apareciendo dos campanas. Las recogeremos y las situaremos en el campanario que tenemos justo en lo alto del edificio colindante, el mismo por el que pasamos una y otra vez, soltando la primera de ellas en uno de los huecos y al coger la segunda, aparecerá en el otro. Automáticamente, se abrirá la tumba y accederemos al interior de la misma. Hay dos pantallas que se repiten en todos los sentidos y hay que tener mucho cuidado, pues lograremos perdernos del todo.

Dentro de ellas hay varias sacas de dinero repartidas cada una en varias estancias. Cogiendo las necesarias y en este orden: la primera saca en la primera pantalla, tal como se entra. Desde la misma posición, pasamos a la siguiente y vamos a por otro botín. La idea es hacerlo de una manera cíclica y rotatoria hasta conseguir que en la parte superior de una de ellas nos aparezca reflectantes las tres figuras de los objetos que previamente habíamos recogido.

Pero antes de subir por la escalera hasta el interior de la cárcel, a modo de curiosidad, si os váis al otro extremo, a la misma altura de donde estabais, al salir hacia la siguiente pantalla, véis que habéis vuelto a la pantalla del lago subterráneo. No tiene mayor trascendencia. Al volver sobre vuestros pasos, ya podréis entrar en la cárcel desde el interior mismo.

Ya dentro de la cárcel, veréis varios prisioneros encarcelados. Les haremos un favor liberándolos uno por uno, pasando por delante de las puertas y pulsando el botón de acción. Procurar evitar los disparos que nos envíen soldados apostados tras las ventanas de la derecha y libraros de aquellos oficiales encargados de la custodia de los presos. Cuando hayan sido todos puestos en libertad, como agradecimiento a vuestra valerosa acción, formarán una escalera humana que os permitiría acceder a la siguiente pantalla. De otra forma, no hubiéseis podido continuar.

Las siguientes pantallas no ofrecen demasiada dificultad, con lo que tenemos a nuestra amada doncella a la vista en menos que canta un gallo. Al llegar hasta élla, nos recordará un gesto como cumplimiento de su compromiso: una flor para sellarlo.





Lógicamente, esto significa volver a la pantalla de siempre y recoger el fruto de su amor, regresando hasta ella por el mismo camino de antes, atravesando las catacumbas y lo demás es coser y cantar. Por otro lado, no esperéis un final espectacular. ¿Pensábais que después de las carantoñas y cucamonas harían explosionar fuegos artificiales y todo el mundo gritaría "hurra"? Pues ni eso...

NEIL PARSONS

Poke de vidas infinitas: 53729,0

Links con información acerca de "Zorro" y archivos de descarga:

http://programbytes48k.wordpress.com/2008/12/12/zorro-datasoft-1985/
http://www.rzxarchive.co.uk/z/zorro.rzx
http://www.worldofspectrum.org/infoseekid.cgi?id=0005874

COMENTARIOS

VERA

Perspective Group, 2008

Existen muy pocos géneros tan ligados a una máquina como lo son las videoaventuras respecto de nuestros Spectrums. Este tipo de juego siempre ha sido de los más genuinos y recordados para nuestra máquina, como si existiera una unión intrínseca entre ambos. Vera es la nueva sensación venida del frío que revive este género con sabor tan ochentero.

Nos hallamos frente a una videoaventura en toda regla, referente de este tipo de juego y de los mejorcito para nuestros Spectrums. El juego consiste en ir recolectando y usando objetos en ciertas zonas y lugares, dichas acciones, normalmente, nos ayudarán a ir avanzando en la aventura de una forma progresiva y escalonada. Algunos objetos tienen un uso lógico pero otros tendremos que buscar-le (o imaginar) donde y como los podemos utilizar, se podría decir que el juego consiste en realizar misiones/acciones inter-medias que unas se unen a otras, hasta finalmente poder dar con el resultado deseado. Existen algunos seres del bosque que son completamente inofensivos o solo hacen cosas cuan-do usamos objetos con ellos.

A nivel gráfico está muy cuidado y es espectacular: gráficos muy definidos y bien caracterizados, colorido realmente acertado y agradable.



El juego destila simpatía y su ambientación está muy conseguida. Tanto la presentación como los menús están perfectamente definidos.

En cuanto a la dificultad hay que decir que el juego es MUY difícil, y cuando digo difícil lo digo en serio. Este tipo de juego es complicado de superar y hemos de dedicar muchísimas horas usando los objetos en muchos lados. Además hemos de añadir que al estar en ruso la cosa se torna realmente difícil. Esta gran dificultad es un acierto y también, por otro lado, su talón de Aquiles puesto que, algunos jugadores, pueden agobiarse y desistir del juego.

Aun con esa enorme dificultad hay que reconocer que es un delicia incluso pasear por los escenarios, visitar nuevas zonas, etc... pocos juegos consiguen ser tan atractivos a la vista como este (incluso hasta para el más inepto jugando con él se puede pasar un buen rato). ***mishazx***

Valoración Global: 8,5

http://www.worldofspectrum.org/infoseekid.cgi?id=0021147

ATOM ANT (Hi-Tec, 1990)

Gráficos y programa: Twilight
Sonido: Sean Conran

*Uno de los personajes más populares y carismáticos de **Hannah-Barbera** fue la hormiga atómica. Como no podía ser de otra manera, Hi-Tec se encargó de su paso por los ordenadores domésticos a principios de los años 90, al igual que hizo con la mayoria de los personajes de la compañia americana. Además, en este caso, el concepto de juego cambió un poco respecto a la mayoria de los demás lanzamientos de la casa británica, asi que en este caso no disfrutaremos de un arcade de plataformas, sino de un juego algo distinto, a un estilo **Bomb Jack**, un poco modificado, eso si. Hi-Tec jugó con la personalidad y las caracteristicas del personaje dotando al programa de un desarrollo muy acorde con dicho insecto, que en el juego tiene la habilidad de moverse con plena libertad por la pantalla, volando y caminando. En lo relativo al movimiento del personaje es bueno advertir que cuando nos encontramos volando, la trayectoria de nuestro (o nuestra??) héroe posee cierta inercia, de forma que cuanto más tiempo pulsemos un botón de dirección, más rápido se moverá la hormiga en esa dirección, y lo que al principio parece un poco incómodo, luego se transforma en un arma a nuestro favor para sortear mejor obstaculos y enemigos complicados.*

Toda esta exhibición voladora de la hormiga atómica le debe servir para desactivar las bombas que están colocadas por la pantalla (otra similitud con Bomb Jack), y que conforme vayamos avanzando estarán colocadas de manera más dificil de alcanzar. Además, también tenemos varios enemigos en cada nivel pululando por la pantalla, que irán aumentando en número y complicación a medida que lleguemos a niveles superiores.

*Pues este Atom Ant, como podéis ver, es básicamente un Bomb Jack a pantalla móvil, es decir, con scroll vertical, enemigos con trayectoria fija, bombas que hay que desactivar (que van apareciendo una a una y que hay que desactivar en un orden determinado siempre, por lo que en cada nivel hay que recorrerse de arriba a abajo todo el mapeado, ya que las bombas suelen estar en sitios extremos algunas de ellas) y una hormiga atómica bastante ágil, que al principio puede parecer un pelín lenta, pero que con la práctica y las partidas nos daremos cuenta de que a veces es mejor la precisión que la velocidad. El juego está bien realizado, los gráficos tengo que admitir que me encantan, especialmente porque el estilo monocromo blanco-negro con sprites tipo cartoon en el Spectrum siempre ha sido una de mis debilidades (con máximo exponente en **Garfield** y **Snoopy** de The Edge). Por otra parte, los sprites están bien definidos, y existe cierta variedad en los fondos, que son principalmente edificios y casas, y también existe variedad en los enemigos, aunque ésta no es mucha, por lo menos agrada ver que los personajes van cambiando.*

>>>>

Por eso, y aunque seguramente no sea muy imparcial en este aspecto, gráficamente Atom Ant me parece un juego agradable de ver, donde los gráficos cumplen mas que de sobra, aunque seguro que tambien habra gente que prefiera ver algo de color en la pantalla (especialmente si estamos tratando juegos basados en dibujos animados). En lo demás, el movimiento, que es decisivo en juegos de este estilo tan dinámico, responde a un buen nivel y no pone ninguna traba a la acción, aunque como ya hemos dicho, al principio puede costar un poco hacerse con el manejo perfectamente, pero al poco de jugar este hándicap desaparece. El Scroll tambien luce a un buen nivel.

Músicalmente, tampoco es un juego descuidado, ya que podriamos definirlo como "calidad Hi-tec", y los que hayais jugado a alguno de sus juegos ya sabeis a lo que me refiero: melodia muy al estilo "cartoon", animada y de buen acabado (obra de Sean Conran) en el menú de opciones. Luego, mientras jugamos, suenan los tipicos efectos de sonido, que cumplen, ni más ni menos. Un detalle curioso en los juegos de Hi-Tec es que ninguno de ellos poseía la melodía original de la serie de dibujos animados en el que se basaba, sino que era Sean Conran (y bastante bien por cierto) el encargado de componer las melodías a partir de cero.

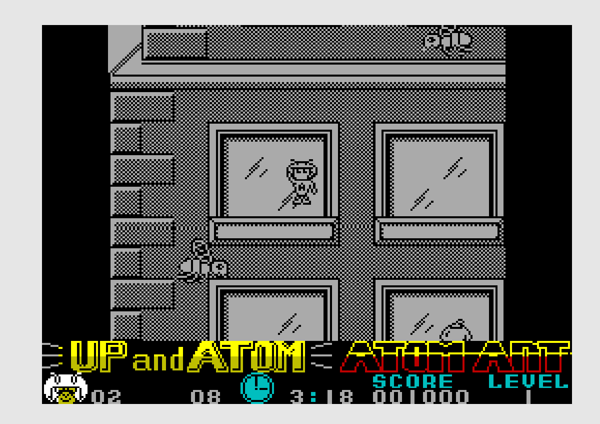
En la parte relativa a la jugabilidad y aspectos adictivos hay que decir que la dificultad del juego lo sitúa en el grupo de los asequibles, pero no fáciles.

Aunque es bastante sencillo superar un nivel o dos, en las fases finales el tema se complica y hay que tener ya adquirida cierta pericia y paciencia, ya que algunas de las situaciones de estos niveles son algo difíciles, aunque nunca desquiciantes ni imposibles. Estamos delante de un juego entretenido, que varía un poco en relación a la mayoria de los lanzamientos de Hi-Tec (algo de agradecer, la verdad) y que mantiene relativamente enganchado al jugador por un tiempo, aunque a veces puede parecer un poco repetitivo y un pelín difícil en los niveles finales.

Recuerdo que en el comentario de MH, la nota final del juego estaba muy cercana al 50%, pero después de haber probado el juego 16 años después por primera vez, tengo la sensación de que no es una nota muy justa, y que debería haber sido algo superior.

Así que ahí lo teneis, coged un Bomb Jack, le ponemos unos pequeños cambios por aqui, y unos detalles por allá, lo hacemos menos trepidante (lo mejor del juego de Elite) y más calculador. En la media de Hí-tec. ***IvanZX***

Valoración Global: 7





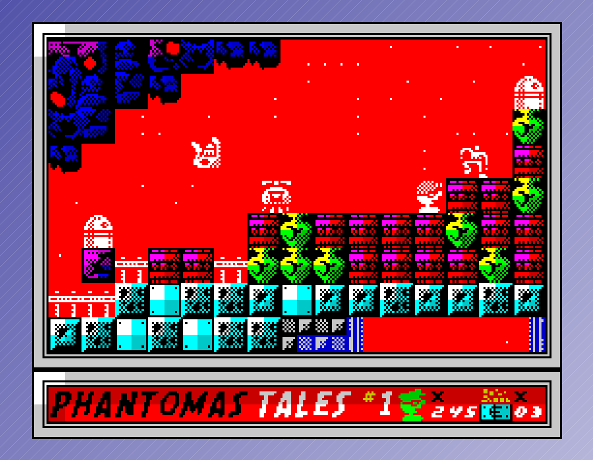
The Mojon Twins

2009

Nathan, Anjuel, Kendrook y WYZ.

48/128k

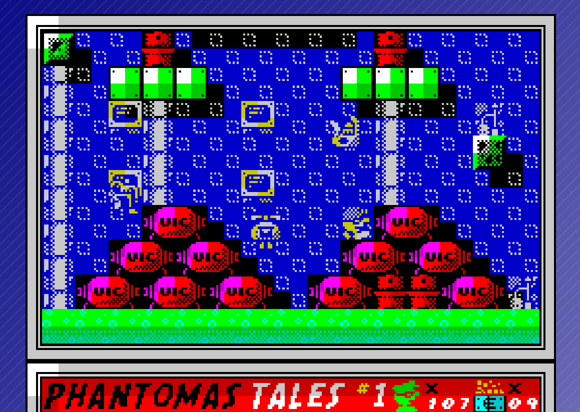
http://www.mojontwins.com/?page_id=171



*Durante siglos, rumores han recorrido las galaxias sobre una secreta civilización, mítica y misteriosa que, afincada en las oscuras y peligrosas cavernas de **Marte**, se dedica a atesorar toneladas del oro cúbico más puro y valioso del Universo. Jamás nadie los ha visto, pero en pocos lugares encontrarás a un incrédulo que niegue su existencia. Simplemente "están ahí", pero "no se dejan ver".*

***Phantomas** llevaba años tras la pista de la misteriosa civilización. Desde que decidió poner fin a la vida de pillaje para la cual fue diseñado tras rebelarse contra sus propias y más profundas raíces, sólo ansiaba construirse un remanso de paz donde descansar, lejos, en el planeta **Novgorod**, entre las montañas de basalto de **Ngailhadam** y los ríos de mercurio **Hyarespone** y **Mithralaigáhn**. Algo tan ambicioso sólo podría costearse con el oro cúbico de **Marte**.*

>>>



*Sería la última misión de **Phantomas** antes de retirarse y tener su bien merecido descanso después de librar al **Universo** de los oscuros conspiradores de **Andrómeda**.*

*Casi por casualidad, **Phantomas** descubrió una comunicación interna mientras reconocía un terreno poco transitado de la superficie de **Marte**. Trazándola laboriosamente pudo encontrar su foco y descubrió, lleno de emoción, la ubicación de una de las minas de extracción de oro cúbico de la tan legendaria civilización desconocida.*

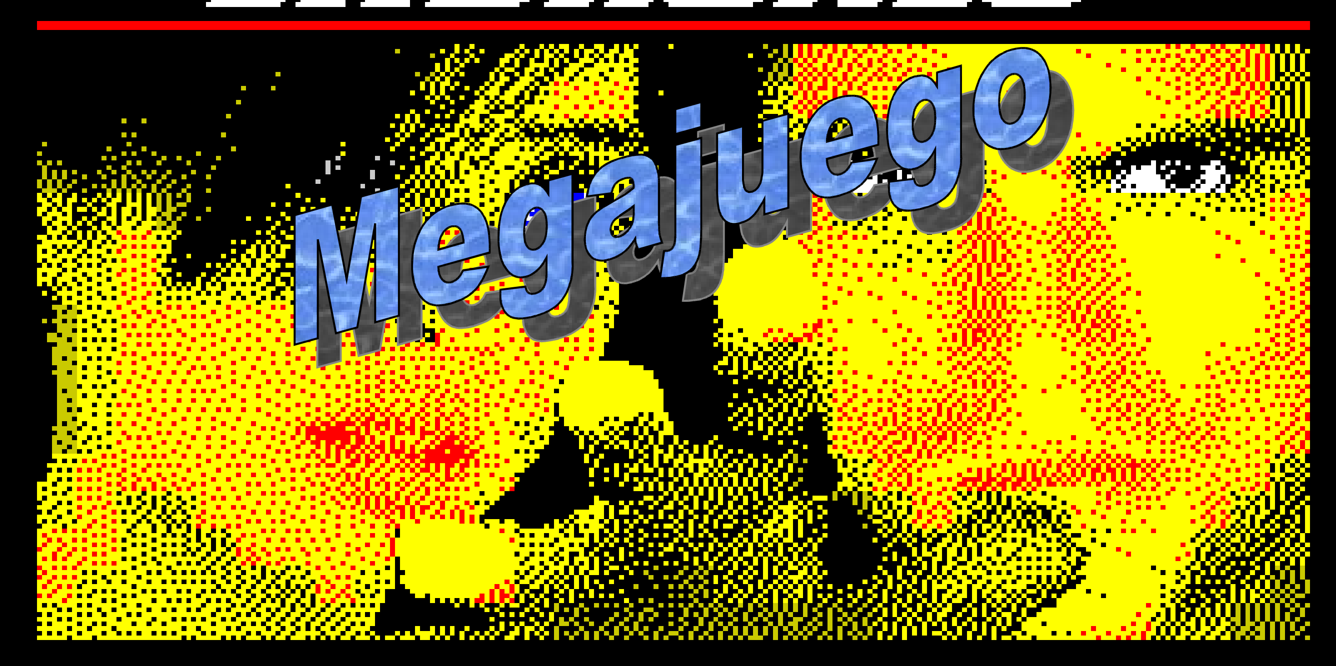
*Así comienza los prolegómenos de esta nueva aventura de Phantomas tras su paso por el **Asteroide Bassard** (ver **Phantomas Saga Infinity**) y en la cual tenemos un "trabajito" que hacer para seguir ganándose la vida. En el planeta rojo tiene que conseguir **reunir 20 contenedores de oro cúbico** de las minas secretas del planeta y la planta de estracción química del mineral, evitando pozos de ácido, pinchos trituradores de rocas, robots de extracción y temibles mineros marcianos de la raza **Kahfre**. Como ya ocurriera en otros juegos de la serie, podemos desplazar a Phantomas a través de las pantallas de izquierda y derecha y realizar dos tipos de salto: Salto ALTO y salto LARGO. Estos saltos serán importantes para desenvolverte mejor por las distintas pantallas y siempre procurando evitar caidas en zonas de peligro como los pozos de ácido y el contacto con los enemigos que nos harán restar puntos de energía.*

*Además de los 20 contenedores de oro, encontraremos **recargas de energía,** que recogidas y usadas, éstas no volverán a aparecer en el mismo lugar, por lo que es necesario consumirlas con cuidado. Éstas serán cada vez más necesarias, especialmente en las zonas más profundas de la planta de extracción química.*

*De jugabilidad está muy bien dotado y estamos ante un programa muy bien depurado y desarrollado y que reune todos los ingredientes para convertirse en un clásico del presente año. No obstante, a pesar del enorme colorido y la genial puesta en escena, ésta puede llegar a ser en ocasiones un inconveniente, al confundirse los sprites con el escenario. La música, tanto en 48k como en 128k, muy trabajada y sobra decir que la de 128k es estupenda y siempre vamos a contar en esta versión con muchos efectos FX y música de fondo de **Anju** y **Nath**. La introducción en forma de cómic, muy buena. En resumidas cuentas, un juego para pasar un largo rato, a la espera de recibir su continuación.* **NeilParsons**

VALORACIÓN GLOBAL: 8,9

ESCUELA DE LADRONES



WORLD XXI SOFT, 2008

Programado por: Ariel Ruiz
Memoria: 48k/128k

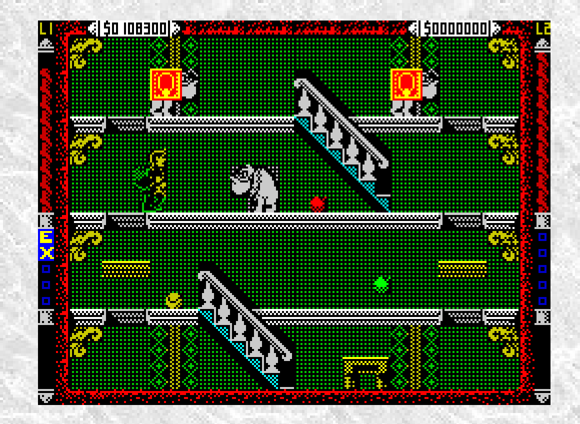
Ariel Ruiz *(aka World XXI Soft Inc) nos presenta el que, a todas luces, es su proyecto más ambicioso publicado hasta la fecha, un desarrollo de más de dos años de duro trabajo que se ha visto reflejado en su nuevo trabajo de título* ***Escuela de Ladrones****.*

La historia nos habla de la peculiar universidad de los amigos de lo ajeno donde nuestros protagonistas, ***Coraje*** *y* ***Bruma****, planean diplomarse con honores. Tras cinco años de fracaso en fracaso, sus maestros, les han dado la última oportunidad de diplomarse: robando por todo el mundo (cual tour turístico) todas las riquezas y joyas que encuentren a su paso. De nosotros depende el éxito o fracaso de ese empeño, ¿estaremos a la altura?*

Para comenzar podemos observar que el juego tiene un claro estilo plataformero pantalla a pantalla, de un tipo de juego muy similar a ***Bubble Booble*** *o* ***Rodland****. (posiblemente el referente más claro sea el Snow Bros de recreativas, sin versión oficial en nuestros Spectrums). En pantalla podemos observar diferentes plataformas, nuestro protagonista, enemigos móviles, etc... y nuestro objetivo es robar todas las riquezas que veamos en cada una de las pantallas. Tras ello debemos eliminar (limpiar) todos los rivales existentes en cada fase arrojándoles todo tipo de elementos que tengamos a mano (pelotas, manzanas, etc.. todas ellas con su peculiar física, efecto de rebote, potencia, etc...). Es interesante observar que los enemigos mueren definitivamente solo si están en la parte absolutamente inferior de la pantalla, por tanto es necesario abatirles del orden de tres veces para que caigan de la parte superior de la pantalla hacía abajo.*

Podemos movernos por el decorado saltando sobre objetos, pasando por detrás de ellos, subiendo escaleras, bajando de una plataforma a otra pulsando la tecla, obviamente, de bajar, etc... De enemigos los hay de diversos tipos: desde móviles con pautas fijas, otros que nos siguen de manera pseudo inteligente del tipo policías, jefes de sector realmente amenazadores o las un poco más anárquicas arañas eso sin contar con la opción dos players, donde podemos tener de enemigo o aliado a un amigo/a jugador. Sin duda hay una buena variedad de elementos llamémoslos molestos e incordiantes.

En general, el programa, nos deja libertad para movernos por la pantalla y tampoco nos impone la presión acostumbrada en este tipo de juegos, del tipo tiempo limitado o similar eso, por tanto, nos deja un poco más de margen para planear como lanzar los disparos e incluso realizar una mini estrategia para superar la escena. Por otro lado, nuestra vida, viene reflejada en la parte superior izquierda de la pantalla, mediante una barra de energía. Perder toda ella supone el triste final.



Gráficos: *En el tema gráfico el juego tiene un gran nivel con bastante colorido, variedad de personajes bien caracterizados, distintos planos de profundidad, etc.... a destacar que el juego consta de dos cargas, siendo la primera una presentación gráfica y sonora bastante cuidada (con elementos digitalizados) y la segunda el juego tal cual. Los gráficos conforme superas fases van ganando en calidad, a mi entender, siendo algunos de ellos, en niveles posteriores, realmente vistosos.*

Sonido: *El tema sonoro es simplemente fantástico,* ***factor 6*** *ha realizado la friolera de unos 16 temas ingame ¡impresionante! y todos ellos consiguen ambientar y mejorar el acabado final del programa. Los FX son correctos y sirven para mejorar aun más si cabe el acabado.*

Movimientos: *El movimiento es bastante fluido a pesar del gran colorido y reacciona perfectamente al teclado (redefinible desde el menú). Posiblemente se agradecería aun más suavidad pero en ningún momento es realmente brusco o feo.*

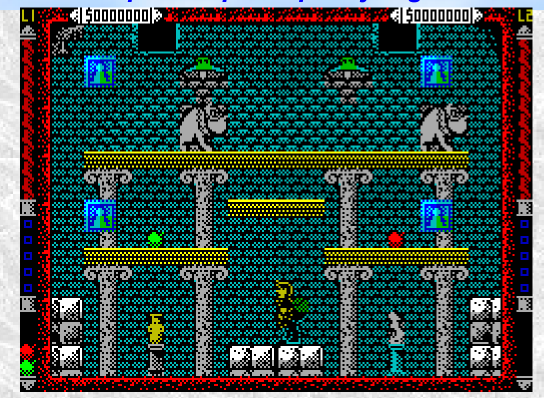
Dificultad: *En cuanto a la dificultad es bastante progresiva en el modo individual (tanto robo como sigilo) y en el cooperativo pues eso, depende de la pericia de ambos jugadores. A nuestro favor diversos elementos que surgen al eliminar enemigos, como por ejemplo el kit médico incremente energía, la bomba que aturde a nuestros rivales como si hubieran recibido un impacto, las letras que si las recoges y formas las palabras* ***EXTRA*** *reponen toda la energía o la letra* ***I*** *que nos hace inmunes por momentos. Uno de los momentos más difíciles del juego es el tema de los jefes, tienen movimientos prefijados pero se hayan protegidos por multitud de secuaces de lo más molesto.*

Adicción. *Para finalizar podemos concluir con que* ***Escuela de ladrones*** *supone un soplo de aire fresco respecto a los juegos de última factura, recuperándonos ese género arcade que tanto nos ha gustado siempre. Es importante destacar que desde el menú podemos cambiar gran cantidad de detalles, desde los diversos modos de juego (mejor y más profundamente explicados en las instrucciones del programa) que le dan mucha más vida al programa, el sonido, cargar expansiones, cambiar idioma, etc.... Este programa, sin duda, supone un salto de calidad bastante notable en los anteriores trabajo de* ***World Siglo XXI*** *y marcan lo que a partir de ahora deben de ser las nuevas producciones de* ***Ariel Ruiz****, esperamos ansiosos.*

IN... *Visualmente muy colorista, adictivo y las músicas son excelentes.* ***OUT....*** *El movimiento sin ser brusco no es 100% suave, las digitalizaciones pueden no gustar demasiado a parte del publico y el que sea de pago limita su distribución respecto a un juego freeware.* ***TONY BRAZIL.***

Valoración Global: 8

http://ws.vtrbandaancha.net/Es/EDL/EDL.htm
http://compiler.speccy.org/

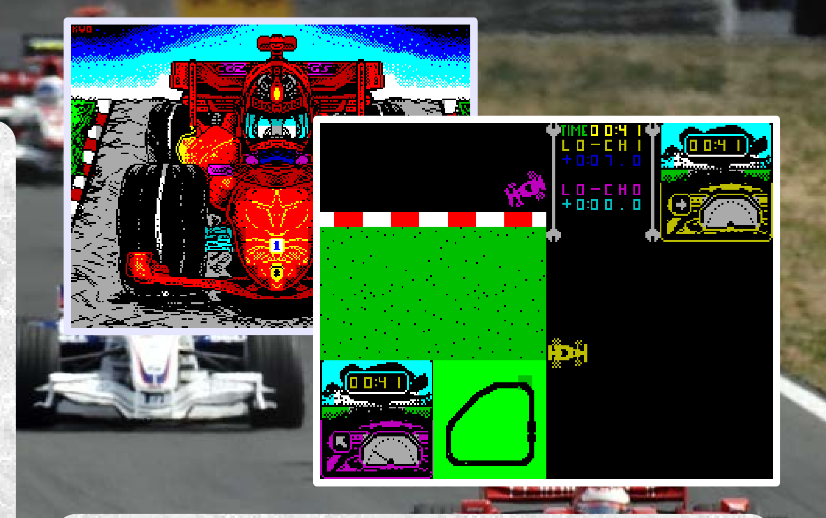


I NEED SPEED

CEZGS - Metalbrain, 2009

*Nos encontramos ante el último juego de la factoría **CEZ Games Studio**, creado por Jaime Tejedor, alias **Metalbrain**, con la gran colaboración de otros maestros de la red que forma **Computer Emuzone**. El título mismo lo dice todo: estamos ante un juego deportivo de carreras de coches y es el último en salir en este sello para el ZX Spectrum, mientras aún queda un último proyecto del sello, el esperado remake del **Capitán Sevilla**, pero eso es otra historia. No hay mucho que decir sobre el argumento, pues es lo de menos.*

*Posee un estupendo **scroll 8×8 pixels** multidireccional, que proporciona sensación de velocidad pero a ratos puede parecer brusco, como también el control de nuestro monoplaza, definible a nuestro gusto o elegido de entre tres opciones predefinidas. El área de juego se ve pequeña para captar los márgenes del circuito que, casi siempre, se salen fuera de la misma, impidiendo en ocasiones predecir en qué dirección giraremos la próxima vez, antes de estrellarnos contra la barrera de protección o salirnos de la pista. Por otro lado, hubiese preferido una pista más estrecha, pero es lo de menos en cualquier caso. Además, aun cuando el área de cada jugador está delimitada por su posición en la pantalla y la colocación de los marcadores, hay ocasiones que puede llegar a confundirnos y no hubiera estado de más una franja vertical separando ambas zonas.*



*No son puntos demasiado negativos, porque a fin de cuentas son apreciaciones que no todo el mundo podrá captar en cuanto se pongan a jugar con él. No obstante, se agradece la buena disposición de los marcadores (marca **Kendroock**), los más que buenos gráficos creados a la par entre Ken, Metalbrain y **Anjuel** y queda por decir las estupendas sintonías acompañando al menú de opciones, realizadas por dos genios de la música por ordenador: **Riskej** y **WYZ**.*

*Particularmente, me ha tenido enganchado durante bastantes minutos el primer dia y he estado jugando con él como veinte veces desde que salió entonces, por lo que queda más que demostrado que es bastante adictivo y gustará, y mucho. **NEIL PARSONS.***

Valoración Global: 8

MARIANO THE DRAGON IN CAPERS IN CITYLAND

CEZGS Silver - The Mojon Twins, 2008

http://www.mojontwins.com/?page_id=70

*Para comenzar la historia del juego nos cuenta la odisea de Mariano, El **Dragón Rosa**, que debe liberar a todos sus amigos de las manos de **Tetrarca Pachin Poi Poi** realizando diversas misiones. Tras realizarlas obtendrá unas llaves que abrirán sus celdas prisión, huyendo de esa manera del planeta **Cityland**.*

Las misiones son las siguientes: Eliminar todos los karakasas (diez) mediante las botas potentes que debemos encontrar. Enviar los planes enemigos a Pachin Poi Poi encontrando el diskette de acceso al ordenador. Pintar los bidones de los hangares de almacenamiento mediante el bote de pintura que, ¡cómo no!, debemos buscar. Encontrar el desmaterializador caza fantasmas y finalmente, y no menos importante, recoger las diez estrellas fugaces. Tras realizar cada una de las misiones obtendremos una llave con la que liberar a todos y cada uno de nuestros amigos: Nanako, Sargento Helmet, Horace, Tracy y Phantomasa respectivamente. Es destacable comentar que cualquiera de las misiones se juega en el orden que deseemos, siendo bastante abierto en su desarrollo y mecánica.

Lo interesante del programa es aprenderte lo antes posible el mapeado, saber donde se encuentran los objetos básicos para realizar las misiones, donde se hayan los lugares donde aplicarlos y, finalmente, en que lugar están cautivos nuestros preciados amigos. Debemos tener en cuenta que los objetos se cogen subiendo sobre ellos y otros, como las estrellas, saltando y/o tocándolos.

A nivel gráfico el programa está bastante cuidado, con gráficos coloristas, bien definidos y cargados de simpatía. La pantalla de presentación, además, parece un agradable homenaje al Humprey de Zigurat. En cuanto al tema sonoro se limita a sonidos eficaces pero simples durante el juego, en cambio en la pantalla antes del inicio nos deleita con una pegadiza, aunque algo corta, melodía en 48k.

Pero, sin duda, el punto más flojo del programa es el movimiento, bastante brusco y poco suave lo cual, a veces, dificulta llegar a ciertos lugares o coger objetos con mucha más facilidad. En cuanto a la dificultad es un juego de relativa complicación y todo depende del orden en que hagamos las misiones. Los enemigos no siguen una pauta fija, siendo un tanto caóticos en sus movimientos. lo cual los hace más imprevisibles. Para algunos esto puede resultar un fallo, en mi caso se vuelve un aliciente de acción notable. Aun así, el nivel de energía inicial (80 puntos de enrgía), está bastante ajustado y nos costará un buen número de partidas terminarlo.

*Mariano es un juego que, aun siendo entretenido y con un acertado gusto a videoaventura, da la sensación de quedarse a medias. De haber sido realizado en **C** o **ASM** podría haber pulido sus defectos y haber llegado a otro nivel. Es loable su intención, pero da la sensación de que podría ser mucho más, y más conociendo otros trabajos de grandísima calidad de su autor como Phantomas Infinity o Phantomasa 2 (por citar algunos). En definitiva, se vuelve un buen tentempié en espera de un nuevo juegazo que, a buen seguro, tendrá su autor en cartera.* **TONY BRAZIL**

Valoración Global: 5,5



PULSOIDS

Tipo: Machacaladrillos
Año: 1988
Programa: Silhouette Software (John F. Cain)
Publicado por: Mastertronic Software.

Que Arkanoid ha sido y sigue siendo uno de los juegos más influyentes de la historia de los videojuegos, eso está claro. Que en la época de los ordenadores de 8 bits hubo mil y una réplicas y copias, eso también está claro. Que hubo muchas que no valieron la pena, no se puede discutir. Pero... no todo fue un "copiar y pegar" sin sentido y sin calidad, sino que hubo algunos casos aprovechables. Y dentro de estos juegos cuyo desarrollo está basado en Arkanoid, que no se pueden considerar como una pérdida de tiempo, encontramos a Pulsoids, que además tiene el aliciente de que nunca fue lanzado en España, y que curiosamente, a pesar de sí haber sido lanzado comercialmente en el Reino Unido, nunca fue comentado por ninguna revista de aquel pais. Así que estamos delante de un juego muy poco familiar, y que ,a pesar de que la idea si que nos es conocida, pasó totalmente desapercibido en nuestro país; posiblemente ni siquiera la facilidad que tenemos ahora con Internet, de poder descargar la mayoria de los juegos existentes para Spectrum haya ayudado a que la gente lo descubra. Hablando ya del juego en si, lo cierto es que no nos podemos detener mucho en ningún aspecto, porque no sobresale en nada, y claro, aún menos en la originalidad. Por eso, prácticamente podemos obviar la historia, que nos sumerge en una lucha espacial, en la que debemos destruir a nuestro enemigo mediante pulsos de energia, que son los que le dan el nombre al juego, y que además definen la principal diferencia entre juegos como Arkanoid y este Pulsoids. En vez de jugar con bolas, lo que hacemos es golpear pulsos o rayos de energia que van destruyendo los escudos enemigos, hasta limpiar la pantalla entera. Estos pulsos de energia en cuanto contactan con un enemigo se multiplican, por lo que es bastante normal que la pantalla se llene de pulsos de energía, lo que nos ayudará a acabar antes con los ladrillos, pero por otro lado hará que debamos estar muy atentos y rápidos.

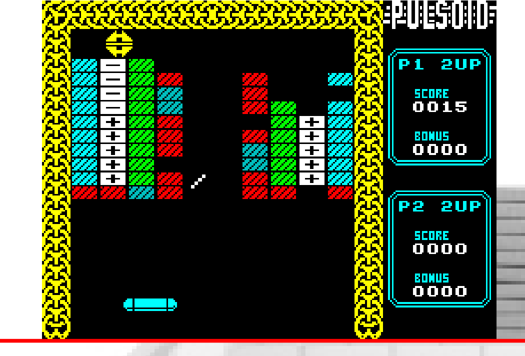
>>>



http://www.worldofspectrum.org/infoseekid.cgi?id=0003936

Por lo demás, como podeis imaginar, tenemos lo típico de este tipo de juegos : los items que proporcionan habilidades especiales, que modifican el comportamiento de la bola, que agrandan la pala, etc, (la variedad no es muy grande en este juego, y prácticamente todos los power ups se han sacado de alguno de los Arkanoids, excepto un par de ellos), los enemigos que pululan por la pantalla y que solamente modifican la trayectoria de los pulsos (a nosotros no nos afectan), y...pare usted de contar. El diseño de las pantallas es más que aceptable, y es algo que se agradece, porque siendo ya de por si un juego nada original, se ve que los programadores pusieron algo de su parte para que no cayeramos en la monotonia. Tampoco se llega al nivel de ninguno de los Arkanoids, pero por lo menos tenemos varios tipos de ladrillos, de enemigos, y de disposición de los ladrillos formando escudos de diferentes formas, con el consiguiente aumento de la dificultad.

Gráficamente está bien, no hay nada especial, buen uso del color, sprites aceptablemente realizados, fondo completamente negro (punto en contra), cierta variedad de enemigos, y nada más que destaque positivamente. Lo mismo que el movimiento, la pala se mueve bien, y solo hay alguna que otra ralentizacion cuando aparecen muchos pulsos y enemigos en pantalla, aunque no afecta casi nada al desarrollo. Los autores tuvieron un poco más de cuidado con el apartado sonoro, y se incluyó una melodía que suena tanto en el menú como en el juego, y que puede cansar a la larga, al repetirse, pero que hay que reconocer que esta medianamente trabajada. Existe la opción de quitar la musica y quedarnos solo con los efectos de sonido, lo cual se agradece tras cierto tiempo jugando.



A la hora de jugar nos encontramos con un programa que realmente no aporta nada nuevo, que se deja jugar, que es medianamente entretenido, que no pasa del montón... y... ya está. Es la misma idea repetida una vez más, realizada con cierto cuidado. Una versión barata del Arkanoid, en la que dirigir la bola adonde queremos no es tan fácil y manejable como en el programa de Imagine, en donde los gráficos, el sonido, los extras están a medio camino, y que por lo menos tiene el reto de ser un juego superable, porque la dificultad está bien ajustada.

¿Porqué este juego no se comentó nunca en las revistas? Posiblemente porque por aquel entonces, año 88, este tipo de juegos estaba muy manido y se consideró que era mejor prestarle atención a muchos otros juegos que salían al mercado. Si hubiera salido en el 84 o 85 hubiera sido un éxito pero... Bueno, para acabar, que si te gustan los machacaladrillos puede que te entretenga durante cierto tiempo en frente de la pantalla, ya que por momentos es adictivo y te mantiene en alerta al tener un ritmo rápido, pero si no te gusta este tipo de juegos, pues mejor no lo pruebes. ***IvanZX***

Valoración Global: 6.5

N.E.X.O.R (Design Design/1986)
Programa: Graham Stafford & David Fish
Tipo: Videoaventura
Memoria: 48k/128k
Descargable desde WOS

En esta ocasión me dispongo revisar uno de esos juegos que, misteriosamente, no salió comentado en las sagradas páginas de nuestra revista más mítica y decisiva, la **Microhobby***. Una cosa extraña ya que, en aquella época, los juegos Filmation eran casi siempre bien recibidos tanto por los usuarios como por la crítica. La historia nos cuenta como un solo hombre debe defender el arma definitiva Nemisis de las fuerzas robóticas de Andrómeda y para ello debe, lo primero, encontrar todas las partes de dicha arma y luego buscar la salida de la zona. Todo ello, además, lo debemos realizar en un tipo limitado (unas tres horas).*

Pasando ya al juego podemos observar dos detalles, el primero es que el menú ya nos recuerda poderosamente a otro Filmation de gran y grato recuerdo llamado Nosferatu y lo segundo es que, una vez entramos al juego en si, se reafirma nuestra impresión: el juego es visualmente muy parecido a Nosferatu y, claro está, sus autores son los mismos Design Design/Graham Stafford, David Fish (aunque la distribuidora del programa sea distinta).

Una vez nos ponemos a jugar ya observamos alguna substancial diferencia, mientras que el clásico de Piranha es una videoaventura con todas las de la ley este N.e.x.o.r busca otra vuelta de tuerca y se asemeja en estilo de juego a los arcade-aventura más recordados de siempre: Batman, Knight Lore o Head Over Heels. Con sus habituales saltos ajustados, suelos que se rompen al pasarlos, bloques que suben y bajan por la pantalla, ene migos móviles o inmóviles que nos eliminan, etc.. todos los elementos característicos de este tipo de juego.

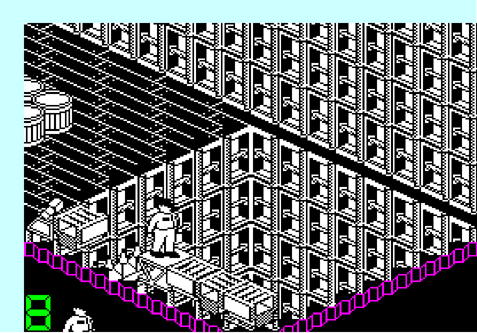
El dilema, la cuestión, etc,,, es saber como ha quedado semejante unión de estética / estilo de juego en un mismo programa y dilu- cidar si nos encontramos ante un clásico, un buen juego o una oportunidad echada a perder.

Si tuviéramos que valorar el juego por su aspecto visual la verdad es que saldría muy bien parado, con unos gráficos tridimensionales isométricos bien definidos pero, por otra lado, un tanto fríos. Las paredes se antojan bastante recargadas comparadas con otros Filmations. Hay detalles curiosos como, por ejemplo, ver la cabeza de margaret tatcher (la política inglesa de aquella época) como enemigo. En cuanto al tema sonoro el juego es prácticamente mudo, profundamente escaso y casi es como no tener sonido alguno. No hubiera costado poner alguna melodía en el menú o algún leve efecto / sonido FX.

Por otro lado el programa reacciona correctamente al teclado y/o joysticks. Al contrario de Nosferatu en este juego un salto ajustado es vital para sobrevivir y se agradece una pronta y adecuada respuesta pero, por desgracia, como muchos Filmations el movimiento se enlentece debido a la sobrecarga de elementos en pantalla. En cuanto a la dificultad disponemos de ocho vidas a todas luces insuficientes dado el gran numero de pantallas y la dificicultad de algunas de ellas. Por otro lado las grandes caídas no nos fulminan, pero resulta un tanto desconcertante y chocante, en la primera pantalla, que si te lanzas al vació estas cayendo durante un buen número de pantallas (esto resulta frustrante y no debería haberse realizado así).

En resumen podemos concluir que este N.E.X.O.R. es un buen juego pero que se queda a medio camino. En lo visual se muestra a gran nivel pero el movimiento de salto, en especial, resulta demasiado poco estilizado y bastante torpe, no queda natural. Eso unido a un sonido casi inexistente hace que el juego se quede en simplemente bueno cuando podría haber sido mucho más..... aun así entretiene y nos puede hacer pasar un buen rato. **TonyBrazil**

Valoración Global: 6



STRONGHOLD

Autor: Red Triangle
Publicado por: Cronosoft (2007)
Tipo: Videoaventura
Memoria: 48/128K

En esta ocasión vamos a tratar un juego de nueva factura, una novedad que apareció en el año 2007, osea hace muy poco, que proviene del grupo de programación ruso Red Triangle y que fue comercializado por Cronosoft. Se trata del primer juego para Spectrum de Red Triangle, y hay que reconocer que para ser su ópera prima, las perspectivas de futuro (ya han afirmado que están manos a la obra con Stronghold 2) son bastante halagüeñas, y que si siguen mejorando tenemos otra compañía a tener en cuenta para alargar la vida de nuestros Spectrums.

Como siempre vamos a describir un poco el argumento y la historia para entrar en acción; En esta ocasión nos adentramos en el mundo de la magia y la brujería: un hechizero de un recóndito pueblo realiza cada cierto tiempo una serie de experimentos con extrañas criaturas; sin embargo, esta vez el experimento se le ha ido de las manos y las criaturas liberadas han puesto en peligro la vida de todos los aldeanos. Para poder resolver el problema debemos internarnos en el templo e ir recogiendo una serie de objetos que nos ayudarán a reestablecer el orden en la villa. Como veis el juego tiene cierto tinte de rol, y la ambientación, la música y los gráficos tienden hacía esa dirección también..

El juego en sí es una aventura, una videoaventura, con pequeñas gotas de arcade (bastante pocas) y con muchos dosis de inteligencia, de resolver los puzzles que se nos plantean en cada pantalla en forma de recoger objetos que luego deberemos usar cuidadosamente en ciertos puntos de dicha pantalla o de las siguientes. Los pequeños detalles de arcade provienen de que deberemos esquivar ciertos elementos del decorado que acabarían con nuestra vida, asi como de los momentos en los que tenemos que eliminar gárgolas con las pócimas.

El juego está resuelto bastante bien, aunque los gráficos son totalmente monocromos, los sprites están muy bien definidos, y tienen un tamaño medio. Eso sí, hay que decir que no hay mucha variedad, pero lo que hay, está bien diseñado. La acción se lleva a cabo dentro de una pantalla que está superpuesta sobre un fondo con un dibujo de una pirámide rodeada de fuego (que en realidad es la pantalla de presentación del juego), así que verse, se ve cierto color en el juego, solo que la ventana de acción en sí es exclusivamente en blanco y negro. El sonido va en linea con el resto del programa, una melodía en el menú, de tipo mistica, muy acorde con el juego, y efectos de sonido variados y conseguidos a la hora de jugar.

A pesar de no ser lo más importante del juego, el movimiento del héroe es bueno, aunque se aprecia cierto retraso dese que pulsamos la tecla hasta que responde; al principio habrá que acostumbarse a esto, pero después no será ningún problema. Como además el juego no es un arcade en el que la precisión es lo más importante, pues no hay que darle mucha importancia a este detalle. Además tiene el misterio de no saber lo que nos espera en la siguiente pantalla, ya que no hay scroll, sino que cada habitación está separada de las demás por puertas, de forma que conforme vamos avanzando nos va picando la curiosidad de descubrir qué habrá mas allá de la última pantalla a la que conseguimos llegar en la última partida. Y aunque las pantallas estén separadas, los objetos y armas que recogemos sirven para cualquier habitación, así que todo en el juego esta interconectado, no es sólo resolver el puzzle de cada pantalla individualmente sino pensar, administrar y estar hábil. Y todo con un nivel de dificultad que anima a avanzar de forma gradual y superable con la práctica.

Una buena idea, bastante bien llevada a cabo, estamos esperando ya la segunda parte. ***IvanZX***

VALORACIÓN GLOBAL: 7,5





INICIACION

Creación de juegos en BASIC compilado (III)

Controlando la RAM ¡Fuera DATAs!

Tras un montón de tiempo sin vernos por aquí, volvemos con el artículo coñazo (coñazo de escribir, vaya) de turno. Seguimos con los subterfugios previos a la chicha, estableciendo debidamente toda la "teoría" que necesitamos comprender antes de ponernos a hacer nada. Sin embargo, para que no se haga tan tedioso (mi meta es que esta serie de artículos siga siendo un coñazo de escribir, pero no de leer), al final del capítulo veremos una aplicación directa de lo que pretendo explicar hoy.

La idea es librarse de todos los subterfugios que BASIC introdujo para suplir su principal deficiencia (o ventaja, como quiera mirarse): ser un lenguaje interpretado. Muchas de las técnicas más habituales de programación en BASIC se basan en el hecho de que se trata de un lenguaje interpretado. El ejemplo más claro de esto son las sentencias **DATA** y toda la parafernalia que las rodea.

Supongo que sabes el BASIC suficiente (prerrequisito para acceder a esta colección de artículos) para saber cómo funcionan las sentencias **DATA**. Para refrescar la memoria, pondremos un ejemplo muy básico:

```
10 DIM s$(4, 10)
20 RESTORE 100
30 FOR i = 1 TO 4: READ s$ (i): NEXT i
100 DATA "NORTE", "SUR", "ESTE", "OESTE"
```

Fijáos en cómo se opera: Necesitamos tener datos en un array en memoria para poder usarlos. Entonces símplemente los leemos del almacén de datos (situado en la linea donde apunta el **RESTORE**) con **READ** y asignamos estos valores a un array. Todo parece bello y bonito, pero tiene dos principales inconvenientes:

1.- Estamos perdiendo un tiempo precioso en copiar una información que está en RAM a otro sitio de RAM que se accede de forma diferente. Y no es una copia normal y corriente, no. Hay mil kilos de comprobaciones y conversiones redundantes de tipos de datos.

2.- Estamos malgastando memoria a cascoporro: por un lado vamos a tener los literales en los DATA ocupando sitio, y luego los vamos a copiar a la zona de variables, ocupando más sitio.

Este tipo de historias se suele emplear en BASIC interpretado por motivos obvios: la única forma rápida de operar con valores es teniéndolos en variables, y no es viable asignar los valores a estas variables con interminables listas de **LET**s (que tampoco eliminarían el problema de los datos duplicados en memoria).

En BASIC interpretado existe una forma de ahorrar espacio: como cuando hacemos un **SAVE** también se gra

ban las variables, muchos programadores, sobre todo en los primeros tiempos, daban valor a estas variables con **READ**s sobre ristras de **DATA**s, las dejaban definidas, borraban los DATAs y el código para leerlos y grababan todo en cinta. Al cargar, las variables estaban ahí. El problema de esto es que no es viable para compilar y, además, cualquier ***RUN*** o ***CLEAR*** inadvertidos se encargarían de destruir todas las variables.

Nosotros queremos compilar. Todo lo que implique una mayor longitud en el código fuente será un problema. Según el manual de la versión para 128K de ***HiSoft BASIC 1.2*** (la que usamos en estos artículos), el compilador es capaz de compilar programas que ocupen 40Kb. Esto es engañoso: jamás compilará 40Kb de código. Esos 40Kb se refieren a código más variables. Si nuestro juego va a manejar muchos datos, hay que reducir la longitud del código todo lo que se pueda.

La solución es sencilla pero puede resultar un tanto inmanejable, dependiendo de para qué necesitemos nuestros datos: todo pasa por POKEar todos estos datos en memoria, por encima de **RAMTOP**, y luego emplearlos directamente con PEEK. Si nuestros datos son cadenas de texto, como en el ejemplo, estamos frente al caso menos favorable: habría que leer las cadenas carácter a carácter, habría que hacer el "padding" a mano (recordad que en un array de cadenas todos los miembros tienen la misma longitud, definida en la segunda dimensión del array, y si una cadena es más corta se rellena con espacios: esto es "padding") y muchas cosas más. Esto es sólo tedioso, porque leer o escribir a memoria desde un binario compilado es muy rápido. Para no caer en el eterno "pecado" de los tutoriales, que se encargan de explicar el caso más sencillo dejando que el sufrido programador tenga que partirse los cuernos con el complejo, vamos a ilustrar los métodos de acceso y control directos de la memoria construyendo una pequeña conversacional que lea todos sus datos con PEEK, directamente de la RAM sobre RAMTOP.

Organizando

Como es lógico, a medida que bajamos de nivel (acercándonos a cómo funciona realmente la máquina, desatendiendo los subterfugios próximos al lenguaje humano que nos ofrecen los lenguajes de alto nivel), hay más y más cosas que debemos controlar por nosotros mismos. Cuando usamos arrays, es el intérprete de BASIC quien se encarga de manejarlas en memoria: de hacer sitio para ellas, de controlar donde están, etcétera. Como nos vamos a saltar a la torera todo esto, somos nosotros los que tendremos que organizar todo lo relativo a su ubicación, y esto exige que pensemos antes de ponernos a programar.

Lo primero, por tanto, es ver qué necesitamos y diseñar nuestro formato de datos. Recordar que vamos a operar a pelo con bytes, así que tenemos control total sobre todo, con todo lo que esto implica. Hemos dicho que vamos a crear una sencilla aventura conversacional. En realidad vamos a hacer mucho menos que eso, va a ser algo realmente sencillo para ilustrar los métodos.

Si el **Sinclair BASIC** estuviese un poco más estructurado, esto que vamos a hacer sería realmente sencillo (por suerte, el compilador ZX Basic que desarrolla Boriel permite hacer esto "como dios manda", pero no es este compilador el que estamos usando aquí: usamos **HiSoft BASIC** que compila Sinclair BASIC puro y duro, así que vamos a tener que liarnos la manta a la cabeza).

Vamos a definir un tipo de datos que llamaremos "LOCALIZACIÓN". En nuestra localización, vamos a definir una descripción, que podrá tener hasta 256 carácteres, y unas salidas que conectarán con otras localizaciones. Para simplificar, usaremos sólo cuatro posibles direcciones: **NORTE, SUR, ESTE y OESTE**. También vamos a desperdiciar un poco de RAM sólo con el objeto de hacer que esto sea mucho más legible. Una vez que comprendamos bien cómo funciona esto, podremos empepinarnos y optimizar y apretujar todo lo que queramos, pero por el momento vamos a ir por la vía clara.

Lo primero que se hace es definir en papel nuestro tipo de datos, para tener en claro qué campos tiene y cuanto ocupa cada campo. Esto es indispensable a la hora de acceder a los datos, y lo tendremos que usar como referencia. Vamos allá:

Offset	Tamaño	Descripción
0	256	Texto con la descripción de la localidad.
256	1	Salida hacia el NORTE
257	1	Salida hacia el SUR
258	1	Salida hacia el ESTE
259	1	Salida hacia el OESTE
260	1	Conexión NORTE
261	1	Conexión SUR
262	1	Conexión ESTE
263	1	Conexión OESTE

Echemos un vistazo: hemos definido todo lo que necesitamos para describir una localización: la descripción de hasta 256 carácteres (que ocupará, por tanto, 256 bytes), cuatro campos "Salida" de 1 byte, que valdrán 1 si hay salida para esa dirección y 0 en casi contrario, y cuatro campos "Conexión" de 1 byte, que contendrán el valor de la localización con la que conecta cada dirección.

Cada localización tendrá un número, de 0 a 255, ya que hemos definido que las conexiones ocuparán 1 byte y en un byte sólo se puede representar un valor de este rango. Por ejemplo, si en la localización 1 tenemos que su byte de "Salida NORTE" vale 1, entonces habrá que mirar el byte de "Conexión NORTE". Si, por ejemplo, vale 7, significará que si salimpos por el NORTE en la localización 1 iremos a la localización 7.

Si contamos, veremos que cada localización nos ocupará 264 bytes (256 de la "Descripción" más 4 de los campos "Salida" más 4 de los campos "Conexión"). Lo que haremos será almacenar en RAM todas las localizaciones de forma contigua, una detrás de la otra. Si las colocamos a partir de "DIRECCION", veremos que cada localización, en RAM, comenzará en la dirección:

DIRECCION + 264 * NUM_LOCALIZACION

Por ejemplo, si **DIRECCION = 50000**, tendremos que la localización 0 empezará en 50000, y que la localización 10 empezará en 50000 + 264 * 10 = 52640.

Ahora que sabemos dónde se ubicará cada localización, tenemos que ver cómo acceder a los valores de los campos. Para eso hay que recurrir a la tablita que nos hemos hecho, y mirar el "offset". El offset no es más que el "desplazamiento" con respecto al principio del registro. Por ejemplo, si queremos leer si la localización 11 tiene salida por el SUR, tendremos que leer el byte que esté en el offset 257 a partir de la dirección de inicio de la localidad 11, lo cual se calcula muy rápido:

```
salida_al_sur = PEEK (50000 + 264 * 11 + 257)
```

¿Véis? Teniendo las cosas bien ordenadas, no es demasiado complicado operar con la información.

Para leer la descripción lo tendremos un poco más complicado, pero no demasiado: sólo hay que irse a la dirección de inicio de la localización concreta que queramos describir, y leer 256 bytes e irlos imprimiendo. Por ejemplo, para imprimir la descripción de la localización 7 haríamos:

```
FOR i = 1 TO 256: PRINT CHR$ (PEEK (50000 +
264 * 7 + i));: NEXT i
```

Escribiendo los datos en memoria

Pero claro, alguna forma tendrá que haber de pokear esos valores en memoria. Y además está el problema de tenerlos que ubicar en el sitio correcto, para aprovechar la RAM todo lo que podamos. Y aquí vuelve a saltar a la palestra el tema de la organización.

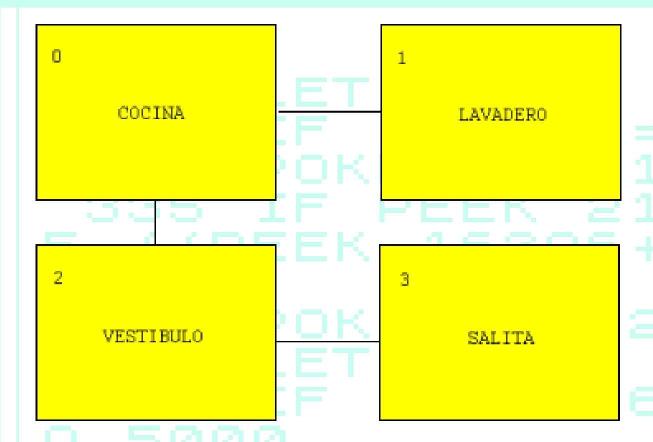
Es indispensable conocer el número de localizaciones que vamos a usar en nuestro juego, porque lo que queremos es colocar nuestros datos al final de la RAM, con el objeto de poner la RAMTOP justo debajo y compilar el BASIC a partir de ahí, como explicamos en el primer capítulo de esta serie.

Para nuestro ejemplo, nos valdrá con 4 localizaciones. Vamos a colocarlas justo debajo de los UDG, que están en 65368. Nuestras 4 localizaciones ocuparán 4*264 = 1056 bytes. Para que nos quepan debajo de los UDG, la dirección de inicio será 65368 - 1056 = 64312. Empezamos definiendo esto:

```
10 LET addr = 64312
```

A partir de aquí, la localización N empezará en **addr + N*264**.

La idea es pokear los datos en memoria y luego salvarlos a cinta como un bloque de bytes (SAVE "datos" CODE addr, 1056). Lo más cómodo es usar DATAs para ello, pokear su contenido en memoria, y luego salvarlo todo. Primero vamos a definir las ristras de datos, y para ello necesitamos elaborarnos un mapita de las cuatro localizaciones y sus conexiones. Vamos en plan sencillo:



Aquí tenemos perfectamente definidas nuestras cuatro localizaciones (numeradas de 0 a 3) y cómo se conectan entre ellas. Por ejemplo, la localización 2, el VESTÍBULO, tiene salidas al NORTE y al ESTE. La del NORTE conecta con la localización 0, y la del ESTE con la 3.

Teniendo esto en claro, lo vamos a pasar todo a DATAs. Os pongo los DATAs, explico qué hay en ellos, y luego comentamos cómo leerlos y POKEarlos en memoria. Empezamos:

```
1000 DATA "Estas en la cocina. Todo esta lleno de mugre y polvo, y puedes distinguir los restos de una opipara cena en proceso de descomposicion."
1010 DATA 0, 1, 1, 0, 0, 2, 1, 0
1020 DATA "Estas en el lavadero. Hay una pila de ropa sucia aun sin lavar, parece que los habitantes de la casa se marcharon con mucha prisa."
1030 DATA 0, 0, 0, 1, 0, 0, 0, 0
1040 DATA "Estas en el vestibulo. Hacia el NORTE puedes ver entreabierta la puerta de la cocina. La puerta de la salita se encuentra hacia el ESTE."
1050 DATA 1, 0, 1, 0, 0, 0, 3, 0
1060 DATA "Estas en una pequeña salita llena de libros viejos y de aberrantes motivos decorativos."
1070 DATA 0, 0, 0, 1, 0, 0, 0, 2
```

Para entender el contenido de estos DATAs, tendrás que echar mano del cuadrito que detalla nuestro modelo de datos. Fíjate que cada localización está definida por una cadena con su descripción y ocho números. Si te fijas en el modelo de datos, tendremos cuatro SALIDAS y cuatro CONEXIONES, en el orden NORTE, SUR, ESTE y OESTE. Si miramos por ejemplo la localización 0, que se define en las lineas 1000 y 1010, podemos ver que estamos usando los números 0, 1, 1, 0, 0, 2, 1, 0. Los cuatro primeros definen las SALIDAS: 0, 1, 1, 0 significa que hacia el NORTE no hay salida (0), hacia el SUR sí (1), hacia el ESTE también (1), y hacia el oeste no (0). Los cuatro siguientes números indican con qué localizaciones se conecta en cada dirección, en el caso de esta dirección valer una. Tenemos 0, 2, 1, 0. Como hacia el NORTE no hay salida, el primer valor puede valer lo que sea. Hemos puesto 0 por poner algo.

Hacia el **SUR** sí hay salida, y tenemos un 2: eso significa que hacia el **SUR** se conecta con la localización 2 (ver el mapa). Hacia el **ESTE** también hay salida, y tenemos un 1: hacia el **ESTE** conectamos con la localización 1. Hacia el **OESTE** no hay salida, por lo que este valor puede valer lo que sea. Hemos puesto 0 igualemente por poner algo. Las otras tres localizaciones se definen de igual forma.

Ahora lo que haremos será un programilla que lea estos DATA, los grabe en memoria, y luego guarde el binario en cinta. Añadimos, por tanto, las siguientes lineas:

```
 10 LET addr = 64312
 20 RESTORE 1000
 30 FOR i = 0 TO 3
 40 REM primero leemos la cadena:
 50 READ d$: LET d$ = d$ + "$"
 60 REM ahora la pokeamos en memoria:
 70 FOR j = 1 TO LEN (d$)
 80 POKE addr + i * 264 + j - 1, CODE (d$
(j))
 90 NEXT j
100 REM ahora leemos las SALIDAS
110 READ n,s,e,o
120 REM y las pokeamos en su offset
correspondiente:
130 POKE addr + i * 264 + 256, n
140 POKE addr + i * 264 + 257, s
150 POKE addr + i * 264 + 258, e
160 POKE addr + i * 264 + 259, o
170 REM ahora leemos las conexiones:
180 READ cn,cs,ce,co
190 REM y las pokeamos en su offset
correspondiente:
200 POKE addr + i * 264 + 260, cn
210 POKE addr + i * 264 + 261, cs
220 POKE addr + i * 264 + 262, ce
230 POKE addr + i * 264 + 264, co
240 REM y hemos terminado, vamos a la siguiente
localizacion
250 NEXT i
260 REM ahora lo tenemos todo en memoria, y lo
grabamos:
270 SAVE "locs" CODE addr, 1056
```

Si seguimos nuestra tablita con el modelo de datos, veremos claramente qué estamos haciendo: estamos leyendo los datos de los DATAs y POKEándolos en memoria, en sus direcciones correspondientes, tal y como hemos explicado antes. Una vez terminado, lo grabaremos todo en cinta.

Lo más complejo aquí puede ser el tema de escribir la cadena en memoria. En primer lugar, tras el **READ** d$, que lee la descripción de los DATAs, estamos añadiendo un carácter "$" al final de la cadena. Esto aquí no sirve para nada, pero nos servirá luego en nuestro juego para saber cuándo termina la descripción y dejar de imprimir.

Acto seguido, iteramos por todos los carácteres de la cadena d$ (***FOR j = 1 TO LEN (d$)***). El carácter j-ésimo de la cadena viene representado por d$(j), por tanto. A la hora de pokear en memoria, no podemos pasarle directamente un carácter, porque POKE espera un valor numérico. Lo mejor es extrar el código ASCII de este carácter, y POKEar este valor. Esto se hace con CODE.

Para saber dónde POKEar, la cuenta es muy sencilla: a partir de addr, sumamos el número de la localización por lo que ocupa la misma: ***i* * *264***. A esto, le sumamos el número del carácter menos 1, ya que el bucle empieza en 1 (porque las cadenas se indexan a parir del 1, y d$(0) no es válido), con lo que tenemos que escribir el carácter j-ésimo en (***addr* + *i* * *264* + *j* - *1***).

El resto de los datos es más sencillo: los leemos con READ y los POKEamos en su offset correcto (mirando la tablita) a partir del inicio de la localización, que, como hemos dicho, es ***addr* + *i* * *264***.

Cuando acaba el bucle i, que itera por las cuatro localizaciones (de 0 a 3), grabamos los datos en cinta. Es interesante, además, salvar este programita, por si queremos cambiar las localizaciones más tarde. En este caso, haremos los cambios en los datas, volveremos a ejecutarlo, y volveremos a grabar el binario con las localizaciones en cinta.

Por lo pronto hacemos un **NEW** para empezar a programar nuestro juego.

Usando estos datos

Teniendo nuestro binario con las localizaciones bien almacenado en cinta, lo primero que tenemos que hacer es proteger la zona de RAM donde vamos a cargarlos estableciendo el RAMTOP justo debajo, para acto seguido cargarlos en esa zona de RAM:

```
CLEAR 64311: LOAD "" CODE 64312
```

Una vez hecho esto, tendremos las localizaciones en RAM y BASIC no las sobrescribirá por mucho que crezca.

El siguiente programa ilustra como usar estos datos en RAM para poder movernos entre las localizaciones:

```
  10 LET addr = 64312: POKE 23658, 0
  20 LET loc = 3: REM empezamos en la salita
  30 GO TO 1000
 100 REM Primero escribimos la descripción:
 110 LET i = locaddr
 120 LET d$ = CHR$(PEEK(i))
 130 IF d$= "$" THEN GOTO 160
 140 PRINT d$;
 150 LET i = i + 1: GOTO 120
 160 PRINT
 170 REM Y ahora escribimos las salidas:
 180 PRINT "SALIDAS: ";
 190 IF PEEK (locaddr + 256) = 1 THEN PRINT
"NORTE ";
 200 IF PEEK (locaddr + 257) = 1 THEN PRINT
"SUR ";
 210 IF PEEK (locaddr + 258) = 1 THEN PRINT
"ESTE ";
 220 IF PEEK (locaddr + 259) = 1 THEN PRINT
"OESTE ";
 230 PRINT
 240 RETURN
1000 LET locaddr = addr + loc * 264
1010 GO SUB 100
1020 LINE INPUT ">>>"; o$
```

>>>

```
1030 IF o$ = "n" AND PEEK (locaddr + 256) =
1 THEN LET loc = PEEK (locaddr + 260): GO TO
1000
1040 IF o$ = "s" AND PEEK (locaddr + 257) =
1 THEN LET loc = PEEK (locaddr + 261): GO TO
1000
1060 IF o$ = "e" AND PEEK (locaddr + 258) =
1 THEN LET loc = PEEK (locaddr + 262): GO TO
1000
1070 IF o$ = "o" AND PEEK (locaddr + 259) =
1 THEN LET loc = PEEK (locaddr + 263): GO TO
1000
1080 PRINT "Direccion incorrecta": GOTO 1020
```

¡Huy, qué susto! Nada, de sustos nada. Si lo miramos despacio y con la tablita del modelo de datos delante, veremos que es muy sencillo.

Nuestro bucle de juego empieza en la linea 1000. Primero calcula la dirección de la localización actual, que guardamos en la variable locaddr. Como ves, sólo suma **264 * loc** a la dirección base de nuestros datos que está en ***addr***. Luego llama a una subrutina en la linea 100 que se encarga de escribir la descripción (iterando a partir de locaddr hasta que se encuentra el carácter $ y escribiendo todo lo demás en pantalla) y luego presentar las salidas (mirando en **locaddr + 256, locaddr + 257**, **locaddr + 258** y **locaddr + 259**, que es donde se especifica esto, ver tablita).

Acto seguido, el bucle del juego pregunta al jugador adónde quiere ir, y dependiendo si escribe "n", "s", "e" u "o" mira si podemos ir en esa dirección (de nuevo mirando en locaddr + 256, locaddr + 257, locaddr + 258 y locaddr + 259, respectivamente). En caso positivo, cambiamos de localización dándole a loc el valor de la CONEXIÓN correspondiente (las conexiones para "n", "s", "e" y "o" están en locaddr + 260, locaddr + 261, locaddr + 262 y locaddr + 263, respectivamente, ver tabla). Hecho esto, volvemos al principio del bucle.

Si compiláis este programa veréis que el binario se ajustará justo por debajo de **RAMTOP** (que es nuestro addr - 1), con lo que podremos grabar todo el bloque de bytes desde el inicio del programa (indicado por HiSoft BASIC) hasta el final de la RAM, como se explicó en el primer programa.

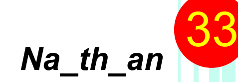
Ahí tendremos todo compactado: binario del programa y datos, en un único bloque de cinta. Si lo ejecutáis (compilado) veréis que funciona rápido y a las mil maravillas. Y nos hemos quitado un chorrazo de código en sentencias **DATA**.

Y con esto y un bizcocho...

Me despido hasta el siguiente capítulo. En el siguiente capítulo emplearemos todo lo que hemos visto en este y el anterior para hacer un sencillo juego tipo Saimazoom, y así aprovecho y explico cómo leer el teclado en condiciones, pasando de INKEY$.

Como siempre, ya sabéis que podéis encontrarme en los foros de **CEZ** o de **Speccy.org**, así que para cualquier consulta allí me tenéis.

¡Que os sea leve!

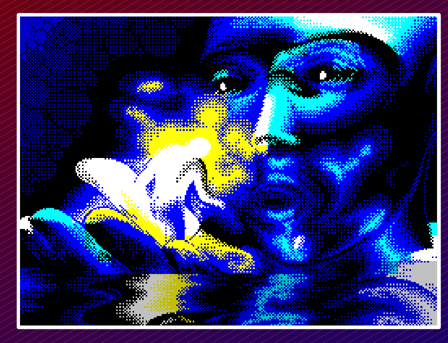


Na_th_an

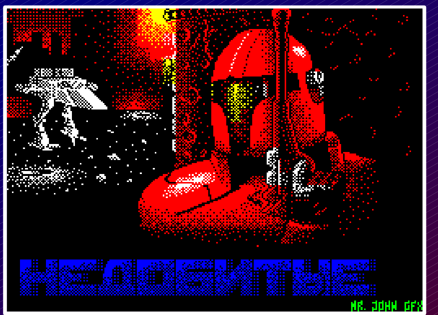
ART GRAFIKA



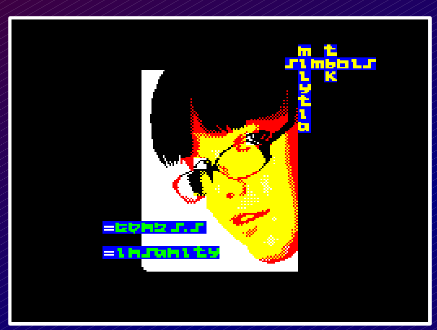
Mushu 2 (Breeze-Fishbone, 2004) >



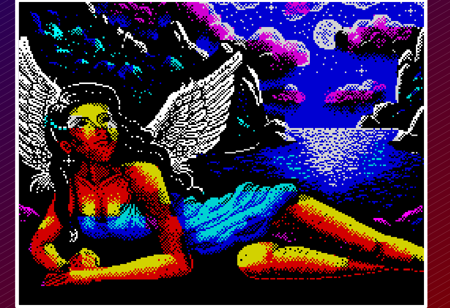
< Blow Hoooo (Deadie - Horrorsoft – CPU)



Nedobity (Mr John / Jurassic Soft) >



< Generation Z issue 5.5 (Simbols, 2009)

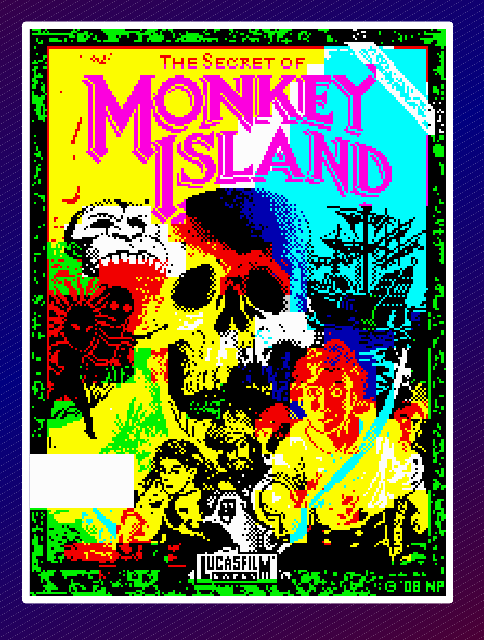


Night Angel (Liza) >

Estas pantallas fueron realizadas por sus correspondientes artistas para diferentes eventos y certámenes de arte gráfico en los últimos años. (Fuente: Demotopia y Raww.org)



< *Shepherd's Truecolor Swings (Riskej)*





< *What Da Bitch (SG of Molytia)*

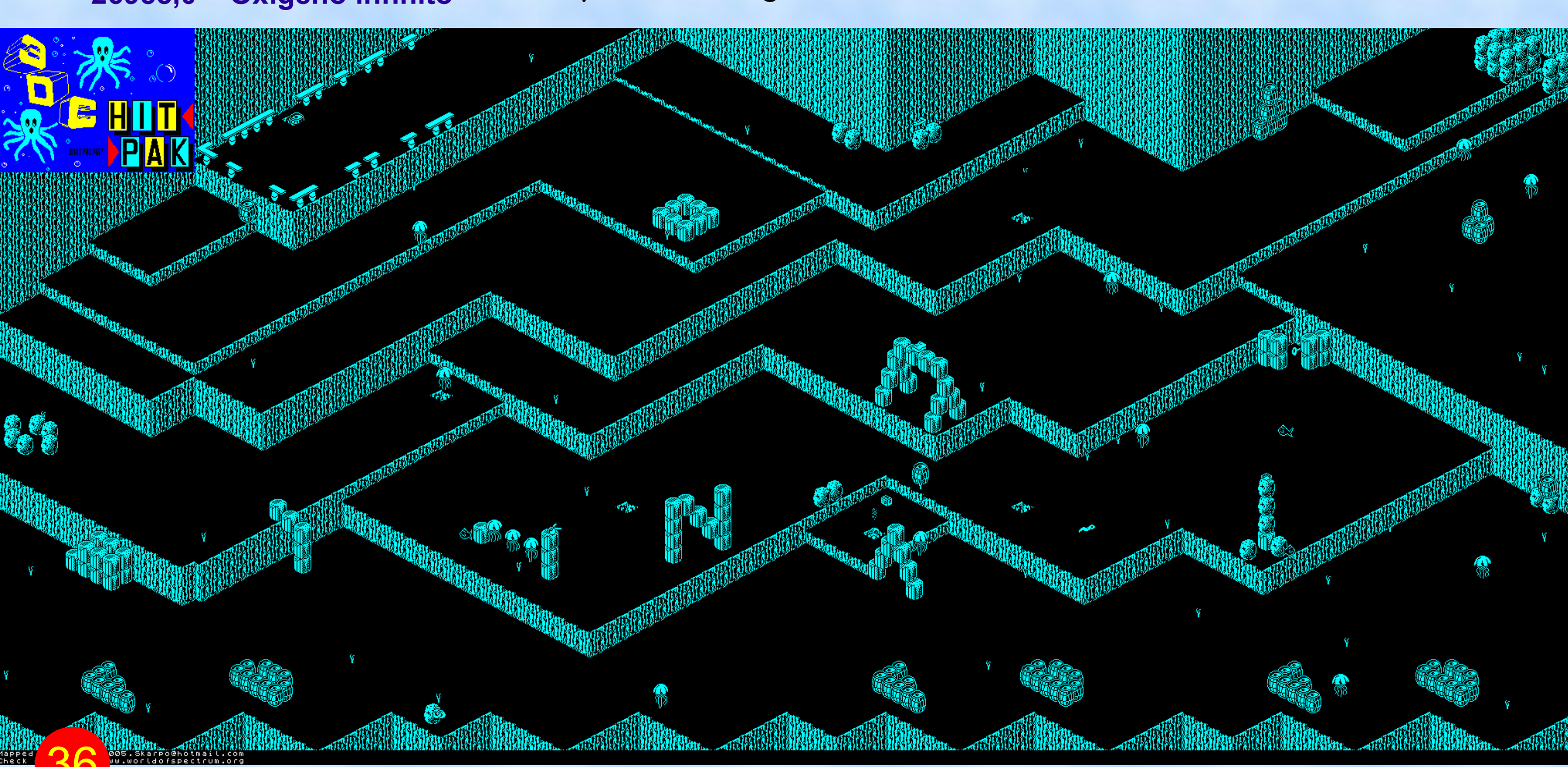
Animo a todos los que tengan alma de artista a realizar pantallas gráficas molonas con el ZX Spectrum y mostrarlas aquí en este espacio de la revista, sin compromiso alguno. Podéis usar las herramientas que consideréis oportunas: Melbourne Draw, OCP Art Studio, Seven-uP, ZX Paintbrush...

Esta pantalla la realicé yo mismo el año pasado, a partir de la carátula original del Monkey Island, primero montando la imagen de dicha carátula de forma que pudiera convertirla con uno de los ya consabidos convertidores de imágenes a formatos de Spectrum y después me pasé un buen tiempo en refinarla hasta dejarla con el aspecto que véis. Por supuesto, desde la pantalla de vuestro monitor tendríais que torcer la cabeza para verla aparecer como un Screen$. La mejor manera de mostrarla es así.

EL MAPA

34298,201 – Inmunidad
34036,0 – Saltos
26955,0 – Oxígeno infinito

En esta edición de la revista, os traigo lo que es el plano donde se desarrolla totalmente la acción del videojuego **3DC** de Elite, incluido en el recopilatorio **Trio Hit-Pak**. Si os cuesta verlo con detalle, este mismo mapa se halla disponible en el archivo de World Of Spectrum y lo podéis descargar desde allí.



ENLACES

http://neilparsons.blogspot.com

http://programbytes48k.wordpress.com

http://www.worldofspectrum.org

http://www.bytemaniacos.com

http://www.webxprs.com/blog

http://radastan.blogspot.com

http://www.speccy.org

http://computeremuzone.com

http://jaimixx.lacoctelera.net/

http://topoxxi.creatuforo.com/index.php





La Portada Alternativa

Una *chorrada* que acometí fotografiando uno de mis Spectrum en un lugar céntrico de mi ciudad. Al final la descarté como portada del presente número, por algún motivo especial y porque pienso que es mejor usar otros elementos con menos riesgo para mi colección particular de ordenadores. Aparte, este Spectrum se cabreó porque quería salir con la **ZX Printer** y no conmigo...


ZX Spectrum
CAPS SHIFT
ENTER
BREAK SPACE
EDIT
CAPS LOCK
TRUE VIDEO
INV. VIDEO
GRAPHICS
DELETE
PLOT
DRAW
REM
RUN
RAND
RETURN
IF
INPUT
POKE
PRINT
NEW
SAVE
DIM
FOR
GOTO
GOSUB
LOAD
LIST
LET
COPY
CLEAR
CONT
CLS
BORDER
NEXT
PAUSE